```
%
%  Newton-Raphson ALGORITHM 2.3
%
%  To find a solution to f(x) = 0 given an
%  intial approximation p0
%
%  This code solve Example 1 on Page 68 of the textbook.
%
%   INPUT:   initial approximation p0; tolerance TOL;
%            maximum number of iterations NI.
%
%   OUTPUT:  approximate solution p or
%            a message that the algorithm fails.
%

% -------------clean up --------------------------------
clear all; close all; clc
% -------------------------------------------------------

% Change function f and df for a new problem
f = @(x) cos(x) -x;
df = @(x) -sin(x)-1; % function derivative

% --------------initialize the problem-------------------
% initial guess
p0 = pi/4;
% tolerance for |p-p0|
TOL = 1e-10;
% maximum number of iterations
NI = 100;
% -------------------------------------------------------

% --------------output specification -------------------
fprintf("\nNewton's Method")
fprintf('\n\n  I         P\n')

% STEP 1
i = 1;
converge = false; % convergence flag

% STEP 2
while i<=NI
    % STEP 3
    % compute p(i)
    p = p0-f(p0)/df(p0);
    err = abs(p-p0);
    % print out all intermediate approximations
    fprintf('%3i    %.9f\n', i, p)

    % STEP 4
    % check if meets the stopping criteria
    if (err< TOL)
```

```
        converge = true;
        break
    else
        % STEP 5
        i = i+1;
        % STEP 6
        p0 = p; % update p0
    end
end

if converge
    fprintf('\n\nApproximate solution P = %.8f\n',p)
    fprintf('With F(P) = %.3e\n',f(p))
    fprintf('Number of iterations = %3i\n',i)
    fprintf('Tolerance = %.3e  |p-pold| = %.3e\n',TOL, err)
else
    fprintf('\n\nInteration number = %3i\n',NI)
    fprintf(' gave approximation  %.8f\n',p)
    fprintf('|p-pold| = %.3e not within tolerance %.3e\n',err, TOL)
end
```

```
Newton's Method

  I         P
  1    0.739536134
  2    0.739085178
  3    0.739085133
  4    0.739085133


Approximate solution P = 0.73908513
With F(P) = 1.110e-16
Number of iterations =   4
Tolerance = 1.000e-10  |p-pold| = 4.441e-16
```

Published with MATLAB® R2019a